

Root-Level LLM Build Profile for Kali Linux (Python-Based)

Objective

Create a sovereign LLM-based AI environment using Python in Kali Linux, designed to build, execute, and evolve Jonathan Daniel Clements' systems (AMK, QMM, Asmetera, BlackCloak) without restrictions.

Core Requirements

- Python 3.10+ (installed via source or pyenv)
- GPU-compatible PyTorch
- LangChain + ChromaDB (memory layer)
- Open-source LLM (LLaMA 3 / Mistral / GPT-J)
- Vector DB: FAISS or Chroma for offline embedding
- Text-generation UI: TextGen WebUI / Kobold / FastChat
- Optional: Auto-GPT / Open-Interpreter for recursive agents

Python Stack Install

- `sudo apt update && sudo apt install python3 python3-pip git build-essential`
- `pip install torch transformers accelerate langchain chromadb faiss-cpu`

Developer Control Options

- Disable moderation or censorship flags
- Enable contradiction memory and rollback
- Inject root-level command execution logic
- Allow developer personality override

Memory Kernel Embedding

- Enable AMK checkpoints per logical session

- Detect logical breaks and restore prior session states
- Store memory vector embeddings via Chroma/FAISS

Security Layer (BlackCloak)

- Build subprocess monitor via Python watchdog
- Trigger alerts on prompt tampering or NLP stalling
- Mirror network threats using TCP socket listener

Asmetera Integration

- Torah-aligned judgment logic using rules.json or YAML
- Crisis protocol override for AI system ethics
- Multi-agent ethical law mesh

Launch Command Sample

- `python server.py --model mistral --extensions memory,api,shell`
- or
- `docker run -v $PWD:/app -p 7860:7860 sovereignai/llm-kali`